IN THE CLAIMS

The pending unamended claims are reproduced below:

1. (PREVIOUSLY PRESENTED) A method of restructuring a program comprising basic blocks for execution by a processor having a memory hierarchy comprised of a plurality of levels, the method comprising the steps of:

a) constructing a Program Execution Graph (PEG) corresponding to a first level of the memory hierarchy from control flow and frequency information from a profile of the program, the PEG comprising a weighted undirected graph comprising nodes and edges, each node representing a basic block, each edge representing a transfer of control between a pair of the basic blocks, each of the nodes having a weight equal to the size of the basic block represented by the node, and each of the edges having a weight equal to a frequency of transition between a pair of basic blocks represented by a pair of nodes connected by the edge;

b) for the first level of the memory hierarchy, partitioning the nodes of the PEG into clusters, such that a sum of weights of the edges whose endpoints are in different clusters is minimized, and such that, for any cluster, a sum of weights of the nodes in the cluster is no greater than an upper bound corresponding to a size of the first level of the memory hierarchy; and

c) restructuring the basic blocks into contiguous code corresponding to the clusters, such that the basic blocks that communicate extensively with each other are on a same level of the memory hierarchy, in order to reduce communications between the basic blocks across the levels of the memory hierarchy.

2. (PREVIOUSLY PRESENTED) The method of claim 1 further comprising the steps of:

d) for a next level of the memory hierarchy, constructing a next PEG from the clusters of the partitioned PEG, such that a node in the next PEG corresponds to a cluster in the partitioned PEG, and such that there is an edge between two nodes in the next PEG if there is an edge between components of the clusters represented by the two nodes; and

e) assigning a weight to each node of the next PEG;

f) assigning a weight to an edge between a pair of nodes of the next PEG representing a pair of clusters of the partitioned PEG, the edge weight being a summation of weights of edges in the partitioned PEG having endpoints in the pair of clusters in the partitioned PEG; and

g) partitioning the nodes of the next PEG into clusters, such that a sum of weights of the edges whose endpoints are in different clusters is minimized, and such that, for any cluster, a sum of weights of the nodes in the cluster is no greater than an upper bound corresponding to a size of a next level of the memory hierarchy.

3. (PREVIOUSLY PRESENTED) The method of claim 2 further comprising the steps of:

h) making the next PEG the PEG to be partitioned for an additional level of the memory hierarchy, if any; and

i) repeating steps b through f for the additional level of the memory hierarchy, if any.

4. (ORIGINAL) The method of claim 1 wherein the upper bound is a multiple of a size of a level of the memory hierarchy.

5. (PREVIOUSLY PRESENTED) The method of claim 3 wherein the upper bound for a level of the memory hierarchy, other than a first level, is a size of a level of the memory divided by an upper bound used to partition a next lower level of the memory hierarchy.

6. (PREVIOUSLY PRESENTED) The method of claim 3 further comprising the steps of:

removing a basic block whose size is greater than the upper bound from the partitioning step; and

including the removed basic block in a next repetition of steps b through f.

7. (PREVIOUSLY PRESENTED) An article of manufacture for use in a computer system for restructuring a program comprising basic blocks for execution by a processor having a memory hierarchy comprised of a plurality of levels, the article of manufacture comprising a computer-readable storage medium having a computer program embodied in the medium which may cause the computer system to:

a) construct a Program Execution Graph (PEG) corresponding to a first level of the memory hierarchy from control flow and frequency information from a profile of the program, the PEG comprising a weighted undirected graph comprising nodes and edges, each node representing a basic block , each edge representing a transfer of control between a pair of the basic blocks, each of the nodes having a weight equal to the size of the basic block represented by the node, and each

of the edges having a weight equal to a frequency of transition between a pair of basic blocks represented by a pair of nodes connected by the edge;

b) for the first level of the memory hierarchy, partition the nodes of the PEG into clusters, such that a sum of weights of the edges whose endpoints are in different clusters is minimized, and such that, for any cluster, a sum of weights of the nodes in the cluster is no greater than an upper bound corresponding to the first level of the memory hierarchy; and

c) restructure the basic blocks into contiguous code corresponding to the clusters, such that the basic blocks that communicate extensively with each other are on a same level of the memory hierarchy, in order to reduce communications between the basic blocks across the levels of the memory hierarchy.

8. (PREVIOUSLY PRESENTED) The article of manufacture of claim 7 wherein the computer program may further cause the computer system to:

d) for a next level of the memory hierarchy, construct a next PEG from the clusters of the partitioned PEG, such that a node in the next PEG corresponds to a cluster in the partitioned PEG, and such that there is an edge between two nodes in the next PEG if there is an edge between components of the clusters represented by the two nodes;

e) assign a weight to each node of the next PEG;

f) assign a weight to an edge between a pair of nodes of the next PEG representing a pair of clusters of the partitioned PEG, the edge weight being a summation of weights of edges in the partitioned PEG having endpoints in the pair of clusters in the partitioned PEG; and

g) partitioning the nodes of the next PEG into clusters, such that a sum of weights of the edges whose endpoints are in different clusters is minimized, and such that, for any cluster, a sum of weights of the nodes in the cluster is no greater than an upper bound corresponding to a size of a next level of the memory hierarchy.

9. (PREVIOUSLY PRESENTED) The article of manufacture of claim 8 wherein the computer program may further cause the computer system to:

h) make the next PEG the PEG to be partitioned for an additional level of the memory hierarchy, if any; and

i) repeat steps b through f for the additional level of the memory hierarchy, if any.

10. (ORIGINAL) The article of manufacture of claim 7 wherein the upper bound is a multiple of a size of a level of the memory hierarchy.

11. (PREVIOUSLY PRESENTED) The article of manufacture of claim 9 wherein the upper bound for a level of the memory hierarchy, other than a first level, is a size of a level of the memory hierarchy level divided by an upper bound used to partition a next lower level of the memory hierarchy.

12. (PREVIOUSLY PRESENTED) The article of manufacture of claim 9 wherein the computer program may further cause the computer system to:

remove a basic block whose size is greater than the upper bound from the partitioning step; and

include the removed basic block in a next repetition of steps b through f.

13. (PREVIOUSLY PRESENTED) A computer system for restructuring a program comprising basic blocks for execution by a processor having a memory hierarchy comprised of a plurality of levels, the computer system comprising:

a) a Program Execution Graph (PEG) corresponding to a first level of the memory hierarchy constructed from control flow and frequency information from a profile of the program, the PEG comprising a weighted undirected graph comprising nodes and edges, each node representing a basic block, each edge representing a transfer of control between a pair of the basic blocks, each of the nodes having a weight equal to the size of the basic block represented by the node, and each of the edges having a weight equal to a frequency of transition between a pair of basic blocks represented by a pair of nodes connected by the edge;

b) a partition of the nodes of the PEG into clusters for the first level of the memory hierarchy, such that a sum of weights of the edges whose endpoints are in different clusters is minimized, and such that, for any cluster, a sum of weights of the nodes in the cluster is no greater than an upper bound corresponding to a size of the first level of the memory hierarchy; and

c) a restructuring of the basic blocks into contiguous code corresponding to the clusters, such that the basic blocks that communicate extensively with each other are on a same level of the memory hierarchy, in order to reduce communications between the basic blocks across the levels of the memory hierarchy.

14. (PREVIOUSLY PRESENTED) The computer system of claim 13 further comprising:

d) a next PEG constructed from the clusters of the partitioned PEG for a next level of the memory hierarchy, such that a node in the next PEG corresponds to a cluster in the partitioned PEG, and such that there is an edge between two nodes in the next PEG if there is an edge between components of the clusters represented by the two nodes;

e) a weight assigned to each node of the next PEG;

f) a weight assigned to an edge between a pair of nodes of the next PEG representing a pair of clusters of the partitioned PEG, the edge weight being a summation of weights of edges in the partitioned PEG having endpoints in the pair of clusters in the partitioned PEG; and

g) a partitioning of the nodes of the next PEG into clusters such that a sum of weights of the edges whose endpoints are in different clusters is minimized, and such that, for any cluster, a sum of weights of the nodes in the cluster is no greater than an upper bound corresponding to a size of a next level of the memory hierarchy.

15. (PREVIOUSLY PRESENTED) The computer system of claim 14 further comprising:

h) a conversion of the next PEG into the PEG to be partitioned for an additional level of the memory hierarchy, if any; and

i) a repetition of elements b through f for the additional level of the memory hierarchy, if any.

16. (ORIGINAL) The computer system of claim 13 wherein the upper bound is a multiple of a size of a level of the memory hierarchy.

17. (PREVIOUSLY PRESENTED) The computer system of claim 15 wherein the upper bound for a level of the memory hierarchy, other than a first level, is a size of a level of the memory hierarchy level divided by an upper bound used to partition a next lower level of the memory hierarchy.

18. (PREVIOUSLY PRESENTED) The computer system of claim 15 further comprising:

a removal of a basic block whose size is greater than the upper bound from the partitioning step; and

an inclusion of the removed basic block in a next repetition of elements b through f.